UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,722	11/09/2000	Stephan J. Jourdan	2207/9800	2194

23838 7590 10/26/2004

KENYON & KENYON
1500 K STREET, N.W., SUITE 700
WASHINGTON, DC 20005

EXAMINER	
O'BRIEN, BARRY J	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 10/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/708,722	JOURDAN ET AL.	
	Examiner	Art Unit	
	Barry J. O'Brien	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment B as received on 8/20/2004.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

4. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 3-8, 12-15 and 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing*, in further view of Johnson, U.S. Patent No. 5,924,092.

7. Regarding claim 1, Patel has taught an instruction segment comprising a plurality of instructions stored in sequential positions of a cache line (see Col.1 line 26 – Col.2 line 15). Patel has not explicitly taught storing the plurality of instructions in sequential positions of a cache line in reverse program order.

8. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

9. Regarding claim 3, Patel in view of Johnson has taught the instruction segment of claim 1, wherein the instruction segment is a trace (see Patel, Col.3 lines 2-12).

10. Regarding claim 4, Patel in view of Johnson has taught the instruction segment of claim 1, wherein the instruction segment is a basic block (see Patel, Col.2 lines 2-5).

11. Regarding claim 5, Patel has taught a segment cache (see "trace cache" of Fig.1) for a front-end system in a processor, comprising a plurality of cache entries to store instructions of

instruction segments (see Col.1 line 26 – Col.2 line 15). Patel has not explicitly taught storing the instructions of instruction segments in reverse program order.

12. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

13. Regarding claim 6, Patel in view of Johnson has taught an apparatus comprising:

a. An instruction cache system (see Patel, "instruction cache" of Fig.1),

b. An instruction segment system, comprising:

I. A fill unit (see Patel, "fill unit" of Fig.1) provided in communication with the instruction cache system, the segment cache of claim 5 included therein (see Patel, Fig.1),

c. A selector (see Patel, "selection logic" of Fig.1) coupled to an output of the instruction cache system and to an output of the segment cache (see Patel, Fig.1).

14. Regarding claim 7, Patel in view of Johnson has taught the apparatus of claim 6, wherein the instruction segment system further comprises a segment predictor (see Patel, "multiple branch predictor" of Fig.1) provided in communication with the segment cache. Here, when the

multiple branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Patel, Col.5 lines 5-29).

15. Regarding claim 8, Patel has taught a method for storing instruction segments in a processor, comprising:

a. Building an instruction segment based on program flow (see Col.1 line 26 – Col.2 line 15),

b. Storing instructions of the instruction segment in a cache entry (see Col.1 line 26 – Col.2 line 15).

16. Patel has not explicitly taught wherein the instructions of the instruction segment are stored in reverse program order.

17. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

18. Regarding claim 12, Patel in view of Johnson has taught the method of claim 8, wherein the instruction segment is a trace (see Patel, Col.3 lines 2-12).

19. Regarding claim 13, Patel in view of Johnson has taught the method of claim 8, wherein the instruction segment is a basic block (see Patel, Col.2 lines 2-5).

20. Regarding claim 14, Patel has taught a processing engine, comprising:

a. A front-end stage to build and store instruction segments, instructions provided therein (see Col.1 line 26 – Col.2 line 15),

b. An execution unit (see "HPS Execution Core" in Fig.1) in communication with the front end stage (see Col.5 line 30 – Col.6 line 4).

21. Patel has not explicitly taught building and storing the instruction segments in reverse program order.

22. However, Johnson has taught the storing of blocks of data in reverse order so that those blocks that were in the first block of a data structure that are frequently accessed and modified will require less moving and fewer modifications after being placed at the last location of data structure, the fewer modifications resulting in improved performance (see Johnson, Col.4 lines 13-24). Because the traces of Patel are indexed, as well as accessed and modified, via their head (first) instructions (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18), one of ordinary skill in the art would have found it obvious to modify the instruction segment of Patel to store the instructions of the instruction trace in reverse order so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved.

23. Regarding claim 15, Patel in view of Johnson has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. An instruction cache system (see Patel, "instruction cache" of Fig.1),

b. An instruction segment system, comprising:

I. A fill unit (see Patel, "fill unit" of Fig. 1) provided in communication with the instruction cache system (see Patel, Fig. 1),

II. A segment cache (see "trace cache" of Fig. 1),

c. A selector (see Patel, "selection logic" of Fig. 1) coupled to an output of the instruction cache system and to an output of the segment cache (see Patel, Fig. 1).

24. Regarding claim 17, Patel in view of Johnson has taught the processing engine of claim 15, wherein the instruction segments are traces (see Patel, Col.3 lines 2-12).

25. Regarding claim 18, Patel in view of Johnson has taught the processing engine of claim 15, wherein the instruction segments are basic blocks (see Patel, Col.2 lines 2-5).

26. Regarding claim 19, Patel in view of Johnson has taught the processing engine of claim 15, wherein the instruction segment cache system further comprises a segment predictor (see Patel, "multiple branch predictor of Fig. 1) provided in communication with the segment cache. Here, when the multiple branch predictor is coupled with the trace cache and mediated by the selection logic, it effectively predicts segments because the basic blocks stored in the trace cache all begin with branches (see Patel, Col.5 lines 5-29).

27. Claims 2, 9-11 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace Packing*, in view of Johnson, U.S. Patent No. 5,924,092, in further view of Peled et al., U.S. Patent No. 6,076,144.

28. Regarding claim 2, Patel in view of Johnson has taught the instruction segment of claim 1, but has not explicitly taught wherein the instruction segment is an extended block.

29. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

30. Regarding claim 9, Patel in view of Johnson have taught the method of claim 9, but have not explicitly taught wherein the method further comprises:

a. Building a second instruction segment based on program flow,

b. If the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment.

31. However, Peled has taught building a second instruction segment based on program flow and subsequently extending the first instruction segment to include the non-overlapping instructions from the second instruction segment if the two segments overlap (see Col.4 lines 13-37) in order to reduce the degree of code redundancy in the trace cache (see Col.1 lines 60-63). One of ordinary skill in the art would have recognized that it is desirable to reduce redundancy within a trace cache so that the cache can be more effectively used and more different traces stored. Therefore, one of ordinary skill in the art would have found it obvious to extend an

existing instruction segment to include non-overlapping instructions from a second instruction segment in order to reduce trace cache redundancy.

32. Regarding claim 10, Patel in view of Johnson in further view of Peled has taught the method of claim 9, but has not explicitly taught wherein the extending comprises storing the non-overlapping instructions in the cache in reverse program order in successive cache positions adjacent to the instructions from the first instruction segment.

33. However, Patel in view of Johnson has taught that instructions in instruction segments are stored in reverse program order (see paragraphs 21-23 above). Because, an extended segment is still an instruction segment, one of ordinary skill in the art would have found it obvious to also store the extended instruction segments in reverse program order.

34. Regarding claim 11, Patel in view of Johnson has taught the method of claim 8, but has not explicitly taught wherein the instruction segment is an extended block.

35. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

36. Regarding claim 16, Patel in view of Johnson has taught the processing engine of claim 15, but has not explicitly taught wherein the instruction segments are extended blocks.

37. However, Peled has taught trace segments which have multiple entry points and a single exit that allow redundant code segments to be eliminated from the trace cache, thereby improving cache utilization (see Peled, Col.1 lines 60-63, Col.4 lines 13-37, and Fig.3). Because the specification has defined an extended block to have multiple entry points and a single exit point (see p.2 of Specification), one of ordinary skill in the art would have found it obvious to modify the instruction segments of Patel to allow for multiple entry points and a single exit so that redundant code segments could be eliminated from the trace cache and performance could be improved.

Response to Arguments

38. Applicant's arguments filed 8/20/2004 have been fully considered but they are not persuasive.

39. On p.6 of the present amendment, the Applicant argues:

> "*Johnson's disclosure does not provide an adequate teaching or suggestion to modify any of Patel's features to arrive at the invention. Johnson nowhere refers to program instructions. Johnson teaches to sort elements of a data array to place the elements most likely to be modifies toward the end of the array. Data arrays commonly store data variables, not program instructions. Johnson's technique allegedly limits the amount of rearrangement that becomes necessary if an element were changed to a new size and re-stored in the array. Such teachings, however, have no application toward the Patel system. Patel's traces store program instructions. Program instructions are not modified. They do not get longer or shorter. They are simply instructions. Accordingly,*

the data arrangement problems noted by Johnson to not occur in Patel's trace-based system."

40. However, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

41. Here, the Applicant is correct in noting that Johnson is directed towards storing data elements in reverse order, while Patel is directed towards storing instruction traces. However, the mere fact that one references is directed towards storing instructions and the other towards storing data does not mean that the references cannot be combined. Using the Johnson reference in the combination rejection does not imply that the entire reference is being incorporated wholly into Patel, but rather that the rejection is using the idea of storing data (instructions are, in fact, data) in reverse order so as to decrease the amount of shifting among other data elements when that data is appended to that Johnson has taught (see above paragraph 8, for example).

42. Furthermore, the Examiner would like to point out that the program instructions of the Applicant's claims are, in fact, data. Nowhere in the claim language are the instructions claimed to have been executed, and thus they are simply data because they are not defined in the claim language as performing any actions, but instead just "exist" (are stored) in cache lines. Thus, the techniques of Johnson can easily be applied to the "data", i.e. instructions, of Patel, and therefore along with the reasons noted above (see above paragraph 8, for example), there is more than sufficient motivation to combine the references.

43. On p.6 of the present amendment, the Applicant further argues:

"Johnson's reverse order idea is based on an expectation that page elements at the 'top' of a page are most frequently modified. As noted above, program instructions within a trace are not modified. When read in context, it becomes clear that Johnson's 'reverse order' statement has no relevance to the claimed subject matter."

44. However, the Applicant has instead read the reference (and its corresponding rejection) out of context. The Applicant is correct in noting that the program instructions within a trace are not modified. However, the Patel reference, as well as the rejection based upon it (see above paragraph 8 for example), is based on the fact that the traces themselves are modifiable, i.e. are able to have instructions (more specifically, pending blocks) added to them (see Patel, Col.3 lines 13-31, describing the merging of a pending block with an existing block), and the fact that they are indexed at their heads (see Patel, Col.4 lines 23-31 and Col.5 lines 12-18 and Col.12 lines 1-22), rather than whether the instructions in the traces are modifiable. Thus, the motivation to combine Johnson with Patel is quite obvious, as both references index a "grouping of data" at a head location, and have the ability to modify the "grouping of data".

Conclusion

45. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

46. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

47. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (571) 272-4171. The examiner can normally be reached on Mon.-Fri. 6:30am-4:00pm, with the exception of the first Friday of every bi-week.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached at (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

48. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
10/20/2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100